Determining the Optimal Number of Clusters with the Clustergram

Joseph K. Fluegemann[1]
University of California Berkeley, San Jose, CA, 95130

Misty D. Davies[2]
NASA Ames Research Center, Moffet Field, CA, 94035

and
Nathan D. Aguirre[3]
New Mexico Institute of Mining and Technology, Raton, New Mexico, 87740

Cluster analysis aids research in many different fields, from business to biology to aerospace. It consists of using statistical techniques to group objects in large sets of data into meaningful classes. However, this process of ordering data points presents much uncertainty because it involves several steps, many of which are subject to researcher judgment as well as inconsistencies depending on the specific data type and research goals. These steps include the method used to cluster the data, the variables on which the cluster analysis will be operating, the number of resulting clusters, and parts of the interpretation process. In most cases, the number of clusters must be guessed or estimated before employing the clustering method. Many remedies have been proposed, but none is unassailable and certainly not for all data types. Thus, the aim of current research for better techniques of determining the number of clusters is generally confined to demonstrating that the new technique excels other methods in performance for several disparate data types. Our research makes use of a new cluster-number-determination technique based on the clustergram: a graph that shows how the number of objects in the cluster and the cluster mean (the ordinate) change with the number of clusters (the abscissa). We use the features of the clustergram to make the best determination of the cluster-number.

I. Introduction

Cluster analysis is a widely used tool in many fields. It uses statistical techniques to organize data in groups, or clusters. The objects or observations (individual data points) in each cluster are grouped such that they are more similar (in general, have less statistical variance), and those in different clusters have maximum variation. Cluster analysis is different than a clustering method because cluster analysis is a process that includes several steps resulting in the formation and interpretation of clusters. The clustering method is a specific algorithm that creates clusters based on the information it is given. According to Milligan, the seven main steps in cluster analysis are: 1) choosing the data elements for the clustering 2) selecting the variables used in the clustering 3) standardizing the variables 4) deciding on similarity/dissimilarity measures 5) choosing the clustering method 6) specifying the number of clusters to be formed 7) interpretation of the results.[1]

However, the main problem with cluster analysis is that much of it is imprecise. Ketchen and Shook note that there is a "stigma" attached to cluster analysis due to the shortage of "underlying theoretical rationale."[2] This shortcoming of cluster analysis is most clearly demonstrated by the frequent need for research intrusion in the process. Because several areas of cluster analysis need significant research judgment, the technique can be considered to lack objective foundation or mathematical basis. These weak points include clustering variable selection, clustering algorithm selection, clustering number determination, and clustering interpretation and validation.

[1] Chemical Engineering Intern, Intelligent Systems Division, 2292 Camrose Avenue, San Jose, CA 95130
[2] Research Computer Engineer, Intelligent Systems Division, M/S 269-1, AIAA Member
[3] Chemical Engineering Intern, Intelligent Systems Division, P.O. Box 3385 Socorro, NM 87801

Determining the number of clusters that will result from the cluster analysis is especially random because the number is usually selected before employing the clustering method. Having to pre-designate the number of groups that will result seems unusual because whatever is sorting something generally has no idea of the number of groups into which to sort the mess until it has tried sorting the given material into different combinations. In many cases, the number of clusters is chosen blindly or with a simple mathematical formula that may or may not apply. With no real rationale for its choice, selecting a cluster-number that is very inaccurate becomes likely, and poor choices will yield terrible results. A number of solutions have been proposed, but all have weaknesses.

Our research consists of constructing a method that will surpass the previous methods for determining the optimal number of clusters for a wide range of data. This new method is based on the clustergram.

Like the popular dendrogram (refer to Figure 1), the clustergram is a tree that shows the movement of cluster observations during the continuous change of groups containing the cluster observations (usually into smaller groups). However, because the dendrogram is strictly hierarchical – each group is continually split – but the clustergram is not – a group or set of observations that is split can come back together, and a new group can receive observations from multiple old groups – it is possible to see how each individual observation moves in a dendrogram but very difficult with a clustergram. As such, although this can be a disadvantage for the clustergram, it allows the clustergram to be used with non-hierarchical clustering algorithms, such as the K-means algorithm, while the dendrogram cannot. Furthermore, the clustergram differs from the dendrogram because, instead of continuing with the present arrangements after each splitting of the data, the clustergram computes each grouping independently of the rest before forming the connections.[3]

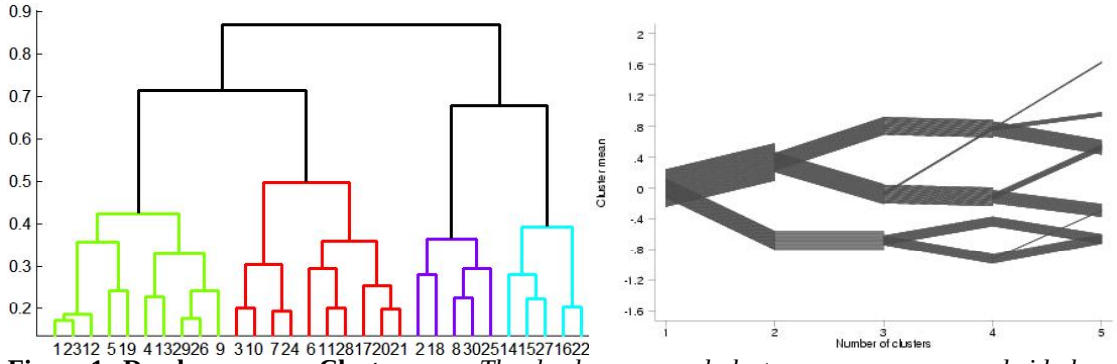

Figure 1. Dendrogram vs. Clustergram. *The dendrogram and clustergram are compared side by side, with a simple example of a dendrogram on the left, and a simple example of a clustergram on the right.*

Specifically, the x-axis of the clustergram is the number of clusters (or some alternate tunable parameter), and the points above each x-value are determined independently of the points above the other x-values. The most general set-up of a clustergram plots the cluster means on the y-axis. However, other possibilities exist, including proportional-to-size, principal components, and weighted means.[4] If the grand mean is meaningful, the clustergram that uses the grand mean displays results that are closest to the actual meaning of the data. As an example of the clustergram's layout (using the cluster means as the y-values), the first three x-coordinates could be 1, 2, and 3, and the selected clustering method would be applied to the data set for these three cluster-numbers; there would be one resulting cluster mean/point for 1 cluster, two resulting cluster means/points labeled '1','2' for 2 clusters, three resulting cluster means/points labeled '1','2','3' for 3 clusters, and so on. Whether the chosen cluster algorithm is run several (or even 20, 30) times for each x-value is optional; clearly, running them many times will ensure that a reasonable clustering result from the algorithm has been chosen, but repeating the process so many times will take space and time and may be rendered unnecessary by the advantageous characteristics of the clustergram.

The main parts of the clustergram are the connecting lines or long parallelograms, which visually display the movement of data points from one cluster to another as the number of clusters increases. The lines/parallelograms vary in width according to the number of data points in both clusters that each line/parallelogram segment connects; these line widths are determined by a proportionality constant that, when multiplied by the number of observations that move between clusters, creates lines that clearly display width differences while avoiding the appearance of clutter as much as possible. The thickness of the cluster line therefore indicates the size of the cluster. A thick clustergram line that remains relativity uniform in its mean (y-axis values) across multiple cluster numbers (x-axis values) represents a strong cluster.

Our research uses the clustergram and the visual information displayed by these connecting line segments/ parallelograms to determine the optimal numbers of clusters into which the clustering algorithm ought to divide the data. Preliminary tests show that, even if there are several instances of poor clustering in the clustergram or even if there are a couple of x-values at which the clustering information taken from the clustering algorithm is completely incorrect, the clustergram still works: because the clustergram measures cluster continuity, even if there are places where the continuity has been destroyed, the overall continuity remains because of the other clusters that were measured correctly. Therefore, the underlying premise of our experiment and results is that, even if the clustering is sub-par, the clustergram will show thick and relatively constant lines sufficiently so that the optimal number of clusters can be seen by counting the number of these wide and even lines.

II. Methods

We created an implementation in MATLAB of a program creating the clustergram. The function we chose for the clustergram (to be displayed on the y-axis) was principal component means (PCM) because grand or weighted means are impractical for many of the multi-dimensional datasets we use, and it is preferable to use some sort of means as opposed to a function like proportional-to-size because the means has direct correlation to the data. The PCM simply projects all data onto the line that maximizes variation; the line is pointed in a direction in which the composite variation among all variables is the greatest.

First, of course, the datasets must be handed to the clustering algorithm, and the clustering algorithm must cluster for a given range of cluster numbers to be displayed on the clustergram. Our experiment uses a variety of clustering algorithms. The clustering algorithms we use include: K-means, Expectation-Maximization (EM) for mixtures of Gaussians, and Density Based Spatial Cluster of Applications with Noise (DBSCAN). K-means was an obvious choice because of its popularity among researchers, owing to its simplicity. However, K-means has the notable weakness of creating clusters with approximately equal size, resulting in the creation or splitting of clusters even when there are none.[1] The EM algorithm is another choice because it is also simple, commonly used, and remedies some of the problems involved with K-means. DBSCAN, as its name suggests, is appropriate for finding clusters based on the density of points, and is thus appropriate for analyzing complex shapes.[5] Like K-means, its popularity stems from its simplicity when compared to other density-based algorithms.

Furthermore, a variety of datasets is provided to each clustering algorithm. The first dataset, which allowed for initial testing of the scripts, is titled BigEasyClustering. BigEasyClustering is specifically suited for EM because it is a mixture of Gaussians having five clear clusters. The clustering plots show the BigEasyClustering points on a two-dimensional layout. Another artificial test-case dataset, the BigDifficultCluster dataset, is meant for DBSCAN because it consists of points composing several odd shapes. Additionally, two common benchmark datasets are the Fisher Iris dataset and the Wisconsin Breast Cancer dataset. Because these datasets are referenced so commonly in cluster analysis literature and because testing our methods on these datasets provides common ground for comparison with other methods, they are included in the research. However, our Fisher Iris dataset is different from the classical one, because ours was enlarged to add 99 points of noise to each original point. This is merely for future experiments testing the number of clusters at a finer scale. The cluster plot displayed for Fisher Iris is the most famous one consisting of three species or clusters that clearly shows one Iris species separated from the other two, but is not as clear on the separation between the other two Iris species. Lastly, a dataset based on E.coli characteristics was included because it is similar to the Fisher Iris dataset, with a few more complications.[6]

Our experiment compares our method for determining the best number of clusters into which to separate the data during the cluster analysis with the Dunn Validity Index. The Dunn Index has been shown to be a reliable validity index for determining the effectiveness of a particular clustering, and thus pointing out the correct number of clusters in a dataset.[7] The Dunn Validity Index checks the within-cluster congruence and between-cluster variance, and assigns a score to that particular clustering based on its measurements (with better clusters receiving higher scores). Thus the cluster-number of the clustering that obtained the highest Dunn score is chosen as the best number of clusters. However, especially when the clustering algorithm is run many times for a long list of cluster numbers in order to apply the Dunn Index, the already innately-space-consuming Dunn Index becomes especially costly. Nevertheless, the Dunn Validity Index continues to be used because it is a reliable method for determining the number of clusters.

Our heuristic uses the clustergram by detecting the longer, thicker, and most constant lines in the plot. The clustergram is useful in this endeavor of determining the optimal number of clusters because it visually shows how the clustering algorithm creates similar clusters for several consecutive cluster-numbers along the x-axis. Therefore, even though specifying a large number of clusters may cause the clustering algorithm to split good clusters into less

well-defined clusters, there will at least exist an interval within the clustergram during which data points were grouped into the same general cluster. The heuristic makes the lines smoother by eliminating the occasional spike from relatively constant lines. It then imposes a cutoff width, or minimum average number of points in a clustergraph line representing a cluster, and calculates the number of these thick and consistent lines as the optimal number of clusters.

III. Results

The results below are grouped by the dataset employed in the cluster analysis. Here there are two datasets: BigEasyClustering and FisherIris. Each section analyzes the performance of each clustering algorithm, a third and final section discusses the results of the Dunn Validity Index for finding the right number of clusters, compared to our method based on the clustergram.

A. BigEasyClustering (Simple Mixture of Gaussians)

As expected, results of clustering the BigEasyClustering set with EM were excellent. With a correct answer of five clusters, the EM algorithm created very accurate groupings when 6 and 8 clusters were specified, but the results were not as accurate when 4 clusters were specified. In other words, once the initial conditions given to the clustering algorithm reached five clusters, except for an anomaly at 10 clusters, there were consistently five large, main clusters. It is interesting that the clustering for 12 clusters was more accurate than for 8 or 10 clusters. Also as expected, the middle cluster was the weakest cluster displayed on the clustergram due to its proximity to the other clusters, resulting in bits and pieces being shaved off into other clusters. Yet, despite any small errors and the mistake that EM made at 10 clusters, the clustergram painted a clear picture of five clusters.

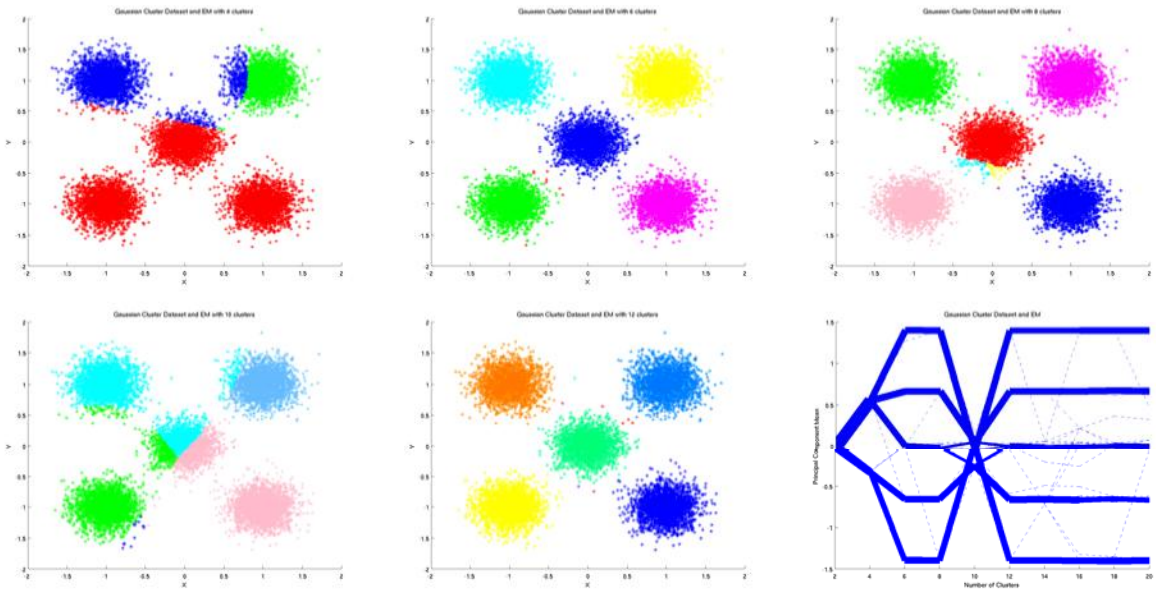

Figure 2. EM with BigEasyClustering. *The first five pictures show how the BigEasyClustering dataset was divided by the EM algorithm for 4, 6, 8, 10, and 12 clusters, respectively, followed by the clustergram.*

DBSCAN was similarly excellent for BigEasyClustering. The lines created by the clustergram working with DBSCAN are long and wide, and the existence of five clusters is unequivocal from this plot. DBSCAN is distinct from the other two algorithms because its initial condition is a minimum number of points in a cluster and not a specific cluster number; as such, DBSCAN has the choice of choosing the number of clusters to create. This meant that for this artificial dataset with inarguably five very distinct clusters, DBSCAN easily picked out the correct clusters, and, from one of the initial minimum cluster observation inputs until the last input, DBSCAN refused to select any number of clusters other than six (five clusters plus a noise cluster). This correct selection for a wide range of inputs is what makes the DBSCAN algorithm and its resulting clustergram so certain of five clusters.

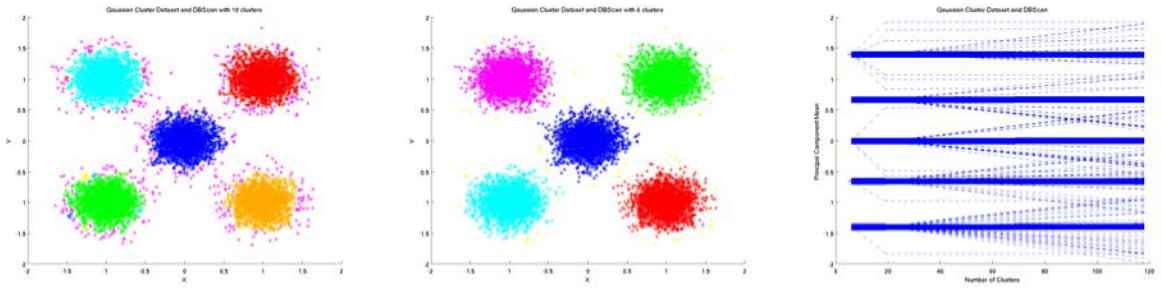

Figure 3. DBSCAN with BigEasyClustering. *The first two pictures show an instance of 19 and 6 clusters, respectively, being formed in the BigEasyClustering dataset by DBSCAN when given a minimum number of cluster objects specification, followed by the clustergram.*

However, K-means was not as clear. Because of the "equal-size" issue, which describes K-means' tendency to create clusters of equal size even when it requires breaking up a good cluster, the clustergram for K-means was not as constant as the other two. Figure 4, especially the picture showing the K-means clustering for 10 clusters, clearly shows the K-means algorithm split certain member of the five clear clusters into nearly equal portions. Nevertheless, the clustergraph does show a general structure with five main branches, and the existence of five clusters is more evident at the beginning of the clustergram, as shown by the five initial bold stubs that split into thinner lines.

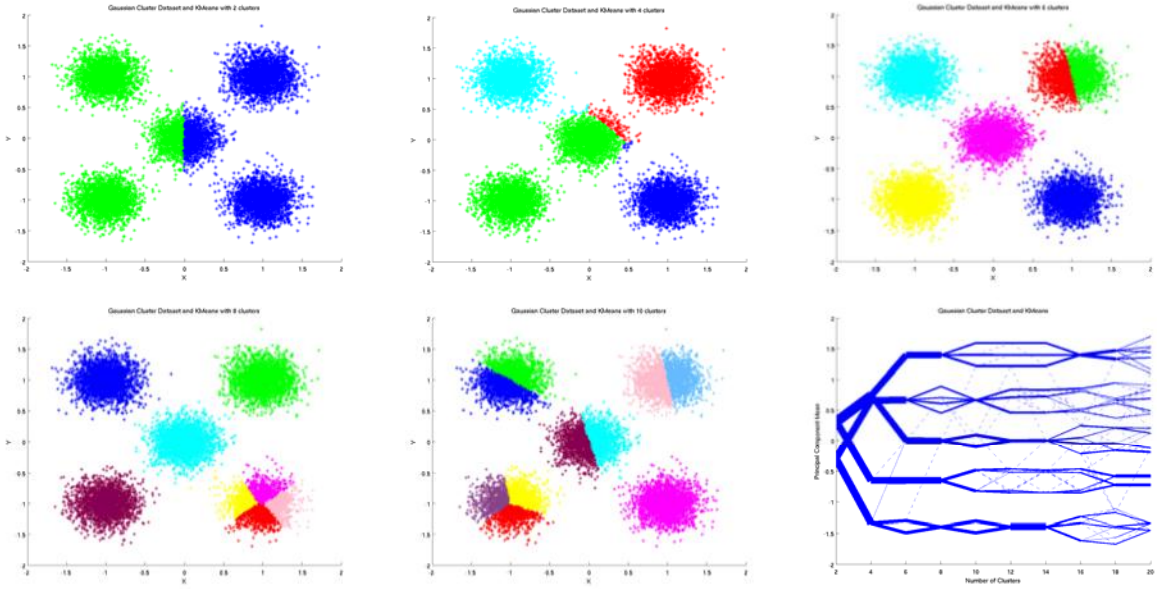

Figure 4. K-means with BigEasyClustering. *The first five pictures show how the BigEasyClustering dataset was divided by the K-means algorithm for 2, 4, 6, 8, and 10 clusters, respectively, followed by the clustergram.*

B. BiggerFisher (Augmented Fisher Iris)

The cluster plots of the way in which the BiggerFisher dataset was divided by K-means were somewhat inaccurate, but not bad overall. The initial division of BiggerFisher into two clusters by K-means can seem unusual at first glance because K-means does not separate the two main groups; however, upon closer examination, K-means is in fact separating two species from another one – the difference is that the two species grouped together are not the two that are closest together in the picture. Later, the K-means algorithm insists on having the Iris species that is most separated from the other two split into two clusters and then dividing the other two Iris species by the remainder of the input. The later plots appear very mottled, with colors scattered all over the diagram. The clustergram for K-means shows two main groups: a rather thick and consistent line along the bottom that is well separated from the rest of the clustergram, which is a bold line initially, but soon splits into five different thin lines, three noticeably thicker than the other two.

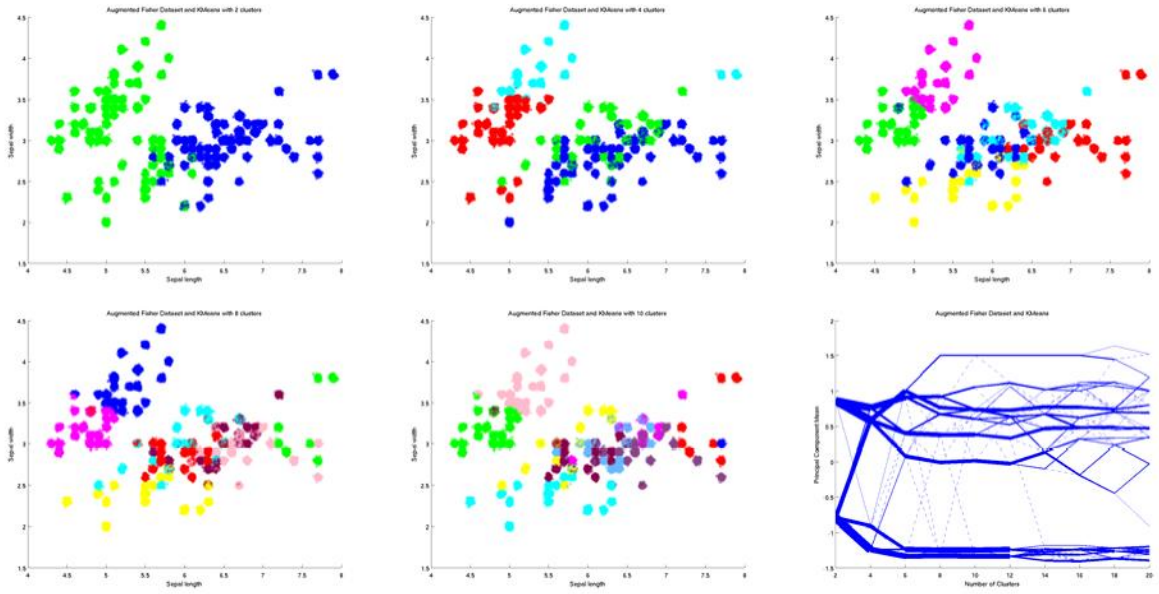

Figure 5. K-means with BiggerFisher. *The first five pictures show how the BiggerFisher dataset was divided by the K-means algorithm for 2, 4, 6, 8, and 10 clusters, respectively, followed by the clustergram.*

The clustergram from Expectation-Maximization with BiggerFisher is similarly to K-means. However, the clustering seems to be a little more accurate in the beginning; for example, when four clusters are selected, the plot contains one cluster that contains the best-separated species of the Fisher Iris dataset. Yet, like K-means, for larger cluster number values, EM creates a panoply of colors seemingly randomly plastered across the plot with the larger of the two groups on the plot receiving more of the colors. This results in a clustergram that looks like the K-means clustergram, including approximately five thinner lines into which the higher group splits.

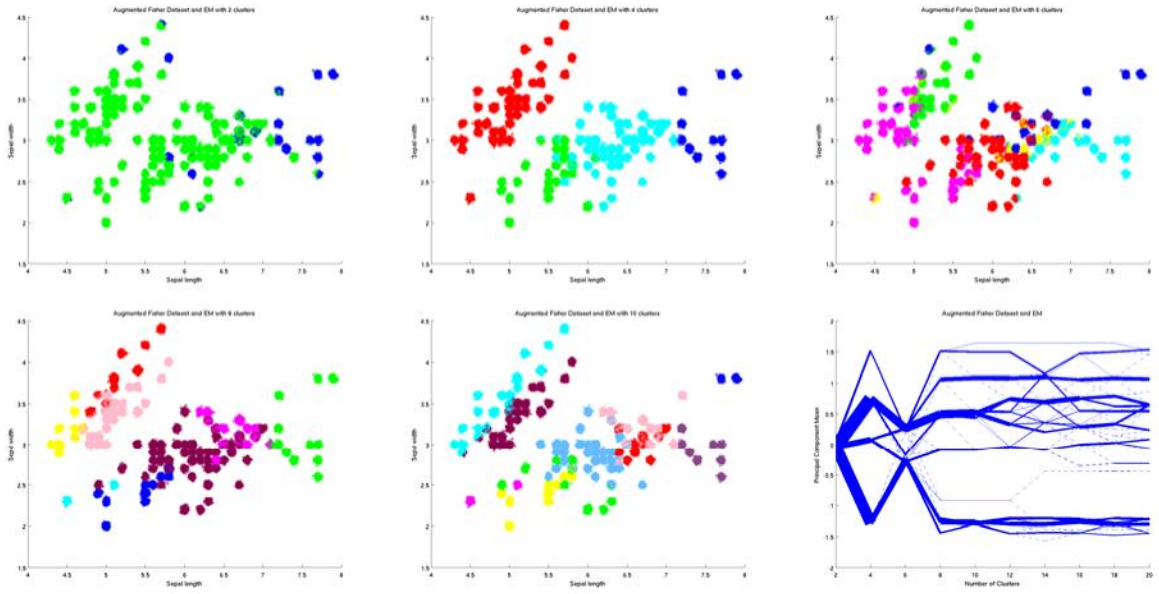

Figure 6. EM with BiggerFisher. *The first five pictures show how the BigEasyClustering dataset was divided by the EM algorithm for 2, 4, 6, 8, and 12 clusters, respectively, followed by the clustergram.*

DBSCAN was no different in creating a variety of different clusters scattered across the plot, some overlapping each other. Unfortunately a large enough minimum number of points for a cluster was not specified, so DBSCAN did not reach a point where it created fewer than fourteen clusters. DBSCAN was able to separate the one species most different from the other two rather quickly – by the time 22 clusters had been reached. However, because DBSCAN is solely density based, DBSCAN separated the larger part of the Fisher Iris plot containing two Iris species by making the center of that part, which contains the largest density of points, and assigning the outer portions to other clusters. The DBSCAN clustergram is similar to the clustergrams of the other two clustering algorithms in terms of the two main divisions, one lower and one higher; however, DBSCAN keeps the upper portion intact as a thick line for a longer period of time before fanning out instead of separating into five or so thin and constant lines.

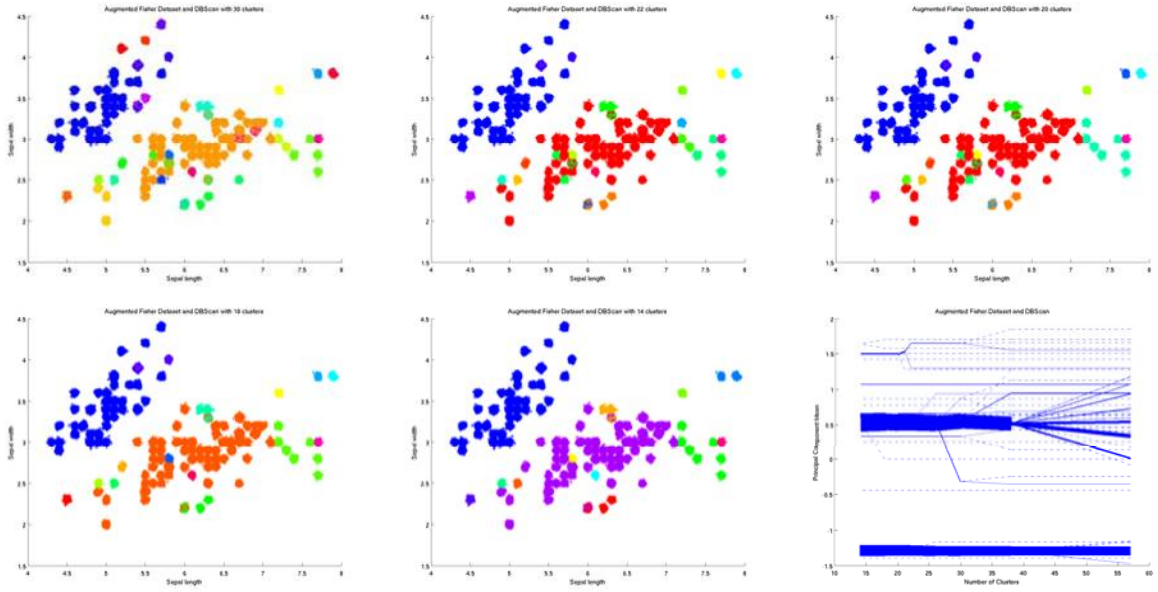

Figure 6. DBSCAN with BiggerFisher. *The first five pictures show instances of 30, 22, 20, 18, and 14 clusters, respectively, being formed in the BiggerFisher dataset by DBSCAN when given a minimum number of cluster objects specification, followed by the clustergram.*

C. Discussion of Dunn Validity Index vs. Clustergram

No pictures are available for the BigDifficultCluster dataset, the Ecoli dataset, or the Wisconsin Breast Cancer datset, but the Dunn Index and clustergram values from the experiment are available, along with the results with the dataset discussed above. They are depicted in Table 1 below. First, for the datasets for which pictures are available and displayed above, the Dunn Index seems to have a slight edge. They were nearly tied for BigEasyClustering, the Gaussian dataset, except for a small mistake by the clustergram when working with K-means. This calls for an improvement in the clustergram program because, even though the K-means clustergram was the most ambiguous of all the clustergrams working with BigEasyClustering, five main clusters for the K-means clustergram are still fairly obvious. The results of the clustering methods with the BiggerFisher dataset were wide and varied. The Dunn Index working with Expectation-Maximization for mixture of Gaussians is obviously the only algorithm that obtained the right answer of 3. However, the Dunn Validity Index was slightly off for K-means and completely off for DBSCAN. The clustergram program was a bit off for all the algorithms working with the augmented Fisher dataset, with EM being the worst by giving an answer (6) that was off the correct answer by 3. It is interesting then, that for the BigDifficultCluster dataset, Dunn Index and clustergram program switch roles when using EM. This time the correct answer was 6 clusters, which the clustergram obtained, but the Dunn Index gave an answer of 3. The other answers from both methods were randomly incorrect. The Ecoli dataset was the one for which no correct answers were attained. Seemingly randomly, the clustergram program working with K-means got the closest to the actual number of clusters: 8. Nevertheless, all results were somewhat distant. With the Wisconsin Breast Cancer data, the Dunn Index performed very well, obtaining the correct answer for all clustering algorithms. While the clustergram program got the right answer in one case, it was off by one in the others.

Dataset (Number Clusters)	Dunn Index	Clustergram
Gaussian Dataset (5)		
K-means	**5**	6
EM	**5**	**5**
DBSCAN	**5**	**5**
Augmented Fisher (3)		
K-means	2	4
EM	**3**	6
DBSCAN	14	1
Big and Difficult (6 +)		
K-means	2	4
EM	3	**6**
DBSCAN	14	1
UCI E.coli (8)		
K-means	2	5
EM	2	2
DBSCAN	2	1
UCI Wisconsin Cancer (2)		
K-means	**2**	**2**
EM	**2**	3
DBSCAN	**2**	1

Table 1. *Dunn Index Values vs. Clustergram Program Values. This table compares the results that the Dunn Validity Index and the Clustergram program obtain for all the datasets and clustering algorithms. The correct results are highlighted.*

IV. Conclusion

The method based on the clustergram has been shown to have some advantages over the Dunn Validity Index. There were a couple of cases in which it obtained results for the correct number of clusters that were closer to the real answers than the Dunn Index. The results demonstrate that the clustergram has an easier time dealing with larger and more complicated datasets where, even if it does not obtain the right answer, it more closely approaches the correct answer than the Dunn Index. The clustergram program at this point needs improvement, but has potential. That the clustergram program would greatly benefit from improvement is evident in at least one example, when the clustergram produced displays the correct number of clusters, and yet the clustergram program determining the optimal number of clusters outputs a result different than what the clustergram shows. With several tweaks, this problem can be remedied. Indeed, the clustergram program has a great deal of potential, not only because it was able to obtain better results than the Dunn Index in a few cases or because it saves space and time, but because the visual evidence is compelling: the clustergram is capable of showing continuity of the main clusters across a range of inputs that points to the optimal number of clusters, even if there are a few poor clusterings, and with some changes, the clustergram program will be able to capitalize on these unique advantages.

There are a few interesting datasets and several other clustering algorithms and cluster-number determination methods that will be used in future work. An image dataset with points corresponding to an image containing well-defined clusters will be a unique dataset that would make for interesting experiments using our method. Furthermore, a very complicated dataset involving university data with a very large number of diverse variables will further test the cluster-number determination method. An additional clustering algorithm is the Fuzzy Clustering algorithm. This clustering method may prove to be useful because of its ability to find clusters even if the "edges" of the clusters are not well-defined. A few additional cluster-number determination methods with which to compare ours include: Calinski and Harabasz, the silhouette method, and the Davies-Boudlin method.[8,9] Calinski and Harabasz's technique has been described as the best-performing index by Milligan and Cooper in their broad survey on the efficacy of various cluster-number determination procedures.[10] The silhouette method calculates a silhouette

coefficient for each point, and the goal of the calculation is for the result to be positive and near zero, while the Davies-Bouldin validation method is another measure of the similarity within cluster to differences between clusters.

Acknowledgments

This research was conducted at NASA Ames Research Center.
Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government.

References

[1]Yan, Mingjin. "Methods of Determining the Number of Clusters in a Data Set and a New Clustering Criterion." Ph.D. Dissertation, Statistics Dept., Virginia Polytechnic Institute and State Univ., Blacksburg, VA, 2005.

[2]Ketchen, D.J., Jr. And Shook, C.L., 1996, The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique, Strategic Management Journal 17(6), 441-459.

[3]Schonlau M. The clustergram: a graph for visualizing hierarchical and non-hierarchical cluster analyses. *The Stata Journal.* 2002;3:316–327.

[4]Schonlau M. Visualizing non-hierarchical and hierarchical cluster analyses with clustergrams. Computational Statistics, 2004.

[5]Mumtaz, K., and Duraiswamy, K. An Analysis on Density Based Clustering of Multi Dimensional Spatial Data, Indian Journal of Computer Science and Engineering, vol. no. 1, pp. 8-12, 2002.

[6]Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[7]J.C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, J. Cybernetics, vol. 3, pp. 32-57, 1973.

[8]P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. 1987. Journal of Computational and Applied Mathematics. 20. 53-65.

[9]D.L. Davies and D.W. Bouldin, "A Cluster Separation Measure," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 1, pp. 224-227, 1979.

[10]Milligan, G.W. and Cooper, M.C., 1985, An Examination of Procedures for Determining the Number of Clusters in a Data Set, Psychometrica 45, 159-179.